Docket No.: L&L-I0225

**MAIL STOP: APPEAL BRIEF-PATENTS**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

| | | | |
|---|---|---|---|
| Applic. No. | : | 10/730,619 | Confirmation No.: 4277 |
| Inventor | : | Burkhard Becker | |
| Filed | : | December 8, 2003 | |
| Title | : | Method and Circuit Configuration for Transmitting Data Between a Processor and a Hardware Arithmetic-Logic Unit | |
| TC/A.U. | : | 2185 | |
| Examiner | : | Yong J. Choe | |
| Customer No. | : | 24131 | |

Hon. Commissioner for Patents
Alexandria, VA 22313-1450

# BRIEF ON APPEAL

Sir:

This is an appeal from the final rejection in the Office action dated March 4, 2008, finally rejecting claims 1 - 3, 5 - 8, 11 - 15, and 18 - 23.

Appellants submit this *Brief on Appeal* including payment in the amount of $510.00 to cover the fee for filing the *Brief on Appeal*.

Real Party in Interest:

This application is assigned to Infineon Technologies AG of München, Germany. The assignment will be submitted for recordation upon the termination of this appeal.

Related Appeals and Interferences:

No related appeals or interference proceedings are currently pending which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

Status of Claims:

Claims 1 - 3, 5 - 8, 11 - 15, and 18 - 23 are rejected and are under appeal. Claim 4 is canceled. Claims 9, 10, 16, and 17 are objected to.

Status of Amendments:

No claims were amended after the final Office action.

Summary of the Claimed Subject Matter:

The subject matter of each independent claim is described in the specification of the instant application. Examples explaining the subject matter defined in each of the independent claims, referring to the specification by page and line numbers, and to the drawings, are given below.

Independent claim 1 reads as follows:

A method [page 15, lines 1-4] for transmitting data of a plurality of data types between a digital processor [page 15, lines 23-24; Fig. 2, DSP] and a hardware arithmetic-logic unit [page 15, line 24; Fig. 2, RW], the method which comprises:

associating the hardware arithmetic-logic unit [page 16, lines 1-3; Fig. 2, RW] with at least one table memory [page 16, lines 1-3; Fig. 2, SP], the hardware arithmetic-logic unit [page 16, lines 1-3; Fig. 2, RW] obtaining data [page 20, lines 15-19] required during a computing operation from the table memory [page 16, lines 1-3; Fig. 2, SP] and/or the hardware arithmetic-logic unit [page 16, lines 1-3; Fig. 2, RW] storing data [page 20, line 24-page 21, line 2] computed during a computing operation in the table memory [page 16, lines 1-3; Fig. 2, SP];

reading and/or writing from the digital processor [page 15, lines 23-24; Fig. 2, DSP] to the table memory [page 16, lines 1-3; Fig. 2, SP] by:

> preselecting [page 17, lines 2-3] a base address in the table memory [page 16, lines 1-3; Fig. 2, SP] dependent on a data type [page 18, lines 16-26] of data to be transmitted;
>
> computing [page 19, lines 11-13] a plurality of addresses according to a prescribed arithmetic computation rule [page 6, lines 19-21] in hardware by taking the preselected base address [page 17, lines 2-5] as a starting point resulting in a computed plurality of addresses; and

accessing the table memory [page 16, lines 1-3; Fig. 2, SP] with the digital processor [page 15, lines 23-24; Fig. 2, DSP] using the computed plurality of addresses for consecutive read access operations and/or consecutive write access operations in the table memory [page 16, line 3; Fig. 2, SP]; and

providing the arithmetic computation rule [page 6, lines 19-21] for computing the plurality of addresses in the table memory [page 16, line 3; Fig. 2, SP] as an incrementation rule or a decrementation rule.

Independent claim 11 reads as follows:

A circuit configuration [page 15, lines 21-24; Fig. 2, CO, AR, C, SP, RW, I/O_M, DSP] for transmitting data of a plurality of data types between a processor [page 15, lines 23-24; Fig. 2, DSP] and a hardware arithmetic-logic unit [page 15, line 24; Fig. 2, RW], comprising:

said processor [page 15, lines 23-24; Fig. 2, DSP] and said hardware arithmetic-logic unit [page 15, line 24; Fig. 2, RW];

at least one table memory [page 16, line 4; Fig. 2, SP], associated with the hardware and arithmetic logic unit [page 15, line 24; Fig. 2, RW], said table memory [page 16, line 4; Fig. 2, SP] for providing data to the hardware and arithmetic logic unit [page 15, line 24; Fig. 2, RW] required for a computing operation of the hardware and arithmetic logic unit [page 15, line 24; Fig. 2, RW], said table memory [page 16, line 4; Fig. 2, SP] for storing data computed in a computing operation of the hardware and arithmetic logic unit [page 15, line 24; Fig. 2, RW];

an input and/or output memory [page 16, line 7; Fig. 2, I/O_M] having a prescribed address [page 16, line 9] used by said processor [page 15, lines 23-24; Fig. 2, DSP] to access [page 16, lines 6-9] said input and/or output memory [page 16, line 7; Fig. 2, I/O_M] for data input/output;

a base address memory device [page 16, line 22; Fig. 2, AR] for storing, for each data type, a base address [page 16, line 23; Fig. 2, BA0, BA1, … Ban] for said table memory [page 17, lines 5-8; Fig. 2, SP]; and

a hardware address computation circuit [page 19, lines 11-13; Fig. 2, C] for, taking the base address [page 7, lines 1-4] as a starting point, applying an arithmetic computation rule [page 6, line 19] to produce a plurality of addresses used by the digital processor [page 15, lines 23-24; Fig. 2, DSP] to consecutively access said table memory [page 16, line 4; Fig. 2, SP];

said arithmetic computation rule [page 6, lines 19-21] being an incrementation rule or a decrementation rule.

## Grounds of Rejection to be Reviewed on Appeal

1. Whether or not claims 1 - 3, 5, 6, 11 - 15, 18 and 20 - 23 are obvious over Hess (US 4,405,980) in view of Tipon et al. (US 5,150,471) under 35 U.S.C. § 103.

2. Whether or not claim 7 is obvious over Hess (US 4,405,980) in view of Tipon et al. (US 5,150,471) and in further view of Stafford et al. (US 3,833,888) under 35 U.S.C. § 103.

3. Whether or not claims 8 and 19 are obvious over Hess (US 4,405,980) in view of Tipon et al. (US 5,150,471) and in further view of Serizawa et al. (US 5,311,523) under 35 U.S.C. § 103.

Argument:

Claims 1 - 3, 5, 6, 11 - 15, 18 and 20 - 23 are not obvious over Hess in view of Tipon et al.

Claims 1 and 11:

Claim 1 includes a step of providing the arithmetic computation rule for computing the plurality of addresses in the table memory as an incrementation rule or a decrementation rule.

Claim 11 includes a hardware address computation circuit for, taking the base address as a starting point, applying an arithmetic computation rule to produce a plurality of addresses used by the digital processor to consecutively access said table memory; said arithmetic computation rule being an incrementation rule or a decrementation rule.

The Examiner has alleged that the ALU would provide a computation rule in order to compute addresses. Column 5, lines 63 to 66 of Hess teach, “The data to be processed, which has been stored in the main memory AKU, is transferred to the arithmetic logic unit ALU via the 4 bit data bus and is processed in accordance with the operation command". From Fig. 1 and its description it becomes apparent that there is a data exchange between the AKU and the ALU. Hess however, does not teach that the ALU uses any type of arithmetic computation rule to compute addresses. Appellant points out that the ALU in Hess simply processes data that is received from the main memory AKU. The ALU taught by Hess does not even compute addresses.

Additionally, claims 1 and 11 specify that the arithmetic computation rule is provided as an incrementation rule or a decrementation rule. In the response to arguments section on pages 9 and 10 of the Office Action mailed on March 4, 2008, the Examiner has stated, "… it is well known to one of ordinary skill in the art that the Arithmetic Logic Unit must provide the arithmetic computation basic rules such as incrementation or decrementation rule in order to compute data, address, etc." In response to this statement, appellant wants to emphasize that even if the ALU is capable of applying an incrementation or decrementation rule, this does not necessarily mean that the ALU actually does apply such a computation rule. Hess does not disclose providing an arithmetic computation rule as an incrementation rule or a decrementation rule. Additionally, as already pointed out, the ALU does not compute addresses.

Tipon et al. do not disclose the above-cited features either. Fig. 1 of Tipon et. al. shows an ALU that provides physical addresses by combining base addresses and offset addresses. This technique does not correspond to the required feature of providing an arithmetic computation rule as an incrementation rule or a decrementation rule.

None of the cited references provide any teaching or suggestion to employ an incrementation rule or a decrementation rule for computing addresses. Appellant asserts that the ALU of Hess does not even compute addresses at all. Therefore, even if one of ordinary skill in the art would have combined the teachings of Hess and Tipon et al., all of the features in claims 1 and 11 would not have been taught or suggested.

Claims 3 and 13:

Claim 3 defines a step of prescribing the plurality of base addresses unalterably in hardware, wherein the plurality of base addresses cannot be processed by the digital processor. Claim 13 specifies that said base address memory device is a read only memory and the plurality of base addresses cannot be processed by the digital processor.

The Examiner has alleged that Tipon et al. teach the features of claims 3 and 13, which have been copied immediately above, and has referred to column 4, lines 5 to 15, which teach a hard-wired base register 18. In this regard, appellant points out that the mere fact that the base register 18 is “hardwired” does not mean that the base addresses are “unalterably prescribed in hardware”.

Column 3, lines 22 to 26 of Tipon et al. teach, “The base address register 18 is loaded by the processor 12 with a preselected, 32 bit base address ... under the control of the processor". If the addresses of the base address register 18 were unalterable in hardware, it would simply not be possible to load the base register 18 with an address. Tipon et al. therefore cannot teach the above cited feature defined in claim 3. A similar argument holds true for claim 13.

Claims 21 and 23:

Claim 21 defines a hardware counter implementing said arithmetic computation rule as an incrementation rule or a decrementation rule. Claim 23 defines performing the step of providing the arithmetic computation rule by providing a hardware counter that implements the incrementation rule or the decrementation rule.

Appellant has already pointed out that the prior art does not disclose applying an incrementation rule or a decrementation rule in the discussion provided with regard to claim 1. With regard to claims 21 and 23 appellant also points out that the prior art does not teach a hardware counter to implement such a computation rule.

## Claim 7 is not obvious over Hess in view of Tipon et al. and in further view of Stafford et al.

Even if Stafford et al. do teach the subject matter that the Examiner has alleged and even if it would have been obvious to combine the teachings of the references, the invention as defined by claim 7 would not have been obtained for the reasons specified above with regard to claim 1 and the deficiencies in the teachings of Hess and Tipon et al.

Claims 8 and 19 are not obvious over Hess in view Tipon et al. and further in view of Serizawa et al.

Even if Serizawa et al. do teach the subject matter that the Examiner has alleged and even if it would have been obvious to combine the teachings of the references, the invention as defined by claims 8 and 19 would not have been obtained for the reasons specified above with regard to claims 1 and 11 and the deficiencies in the teachings of Hess and Tipon et al.

The honorable Board is therefore respectfully urged to reverse the final rejection of the Primary Examiner.

Any fees due should be charged to Deposit Account No. 12-1099 of Lerner Greenberg Stemer LLP.

Respectfully submitted,

/Mark P. Weichselbaum/
Mark P. Weichselbaum
Reg. No. 43,248

/lq
Date: September 8, 2008
Lerner Greenberg Stemer LLP
Post Office Box 2480
Hollywood, Florida 33022-2480
Tel: (954) 925-1100
Fax: (954) 925-1101

Claims Appendix:

1. A method for transmitting data of a plurality of data types between a digital processor and a hardware arithmetic-logic unit, the method which comprises:

associating the hardware arithmetic-logic unit with at least one table memory, the hardware arithmetic-logic unit obtaining data required during a computing operation from the table memory and/or the hardware arithmetic-logic unit storing data computed during a computing operation in the table memory;

reading and/or writing from the digital processor to the table memory by:

> preselecting a base address in the table memory dependent on a data type of data to be transmitted;
>
> computing a plurality of addresses according to a prescribed arithmetic computation rule in hardware by taking the preselected base address as a starting point resulting in a computed plurality of addresses; and
>
> accessing the table memory with the digital processor using the computed plurality of addresses for consecutive read access operations and/or consecutive write access operations in the table memory; and

providing the arithmetic computation rule for computing the plurality of addresses in the table memory as an incrementation rule or a decrementation rule.

2. The method according to claim 1, which further comprises:

storing a plurality of base addresses associated with a plurality of different data types in a base address register, the base address that was preselected being one of the plurality of base addresses; and

performing the step of preselecting the base address by using the processor to set a selection bit associated with the base address.

3. The method according to claim 2, which further comprises:

prescribing the plurality of base addresses unalterably in hardware, wherein the plurality of base addresses cannot be processed by the digital processor.

5. The method according to claim 1, which further comprises:

programming the base address with the digital processor.

6. The method according to claim 1, which further comprises:

with the digital processor, programming at least one information item selected from a group consisting of information relating to a number of data items being written to or read from a plurality of memory subareas associated with the base address, information about a block size of data blocks, information about a decoding rate, and information about utilized convolution polynomials.

7. The method according to claim 1, which further comprises:

providing a first data type of the plurality of data types as soft input values for channel decoding that are intended for a Viterbi decoder hardware arithmetic-logic unit; and

with the digital processor, programming how many soft input values per unit time can be stored in a memory subarea associated with the first data type.

8. The method according to claim 1, which further comprises:

providing a second data type as trace back values computed by a Viterbi decoder hardware arithmetic-logic unit; and

with the digital processor, programming how many states the trace back values need to include.

11. A circuit configuration for transmitting data of a plurality of data types between a processor and a hardware arithmetic-logic unit, comprising:

said processor and said hardware arithmetic-logic unit;

at least one table memory associated with the hardware and arithmetic logic unit, said table memory for providing data to the hardware and arithmetic logic unit required for a computing operation of the hardware and arithmetic logic unit, said table memory for storing data computed in a computing operation of the hardware and arithmetic logic unit;

an input and/or output memory having a prescribed address used by said processor to access said input and/or output memory for data input/output;

a base address memory device for storing, for each data type, a base address for said table memory; and

a hardware address computation circuit for, taking the base address as a starting point, applying an arithmetic computation rule to produce a plurality of addresses used by the digital processor to consecutively access said table memory;

said arithmetic computation rule being an incrementation rule or a decrementation rule.

12. The circuit configuration according to claim 11, wherein:

said base address memory device is an external base address register designed such that in order to select the base address, said processor sets a selection bit associated with the base address.

13. The circuit configuration according to claim 11, wherein said base address memory device is a read only memory and the plurality of base addresses cannot be processed by the digital processor.

14. The circuit configuration according to claim 11, wherein said base address memory device is a rewritable memory that can be programmed by the digital processor.

15. The circuit configuration according to claim 11, further comprising:

a configuration memory;

said table memory including memory subareas; and

said configuration memory for storing information selected from a group consisting of information relating to a number of data items being written to or read from a plurality of said memory subareas associated with the base address, information about a block size of data blocks, information about a decoding rate, and information about utilized convolution polynomials.

18. The circuit configuration according to claim 11, wherein said table memory has a prescribed memory word length.

19. The circuit configuration according to claim 11, wherein:

said hardware arithmetic-logic unit is a Viterbi hardware arithmetic-logic unit.

20. The circuit configuration according to claim 11, wherein:

said hardware arithmetic-logic unit includes an equalizer hardware arithmetic-logic unit and a decoder hardware arithmetic-logic unit;

said processor includes a data transmission connection to said equalizer hardware arithmetic-logic unit; and

said processor includes a data transmission connection to said decoder hardware arithmetic-logic unit.

21. The circuit configuration according to claim 11, further comprising:

a hardware counter implementing said arithmetic computation rule as an incrementation rule or a decrementation rule.

22. The circuit configuration according to claim 11, wherein the plurality of base addresses cannot be processed by said processor.

23. The method according to claim 1, which further comprises:

performing the step of providing the arithmetic computation rule by providing a hardware counter that implements the incrementation rule or the decrementation rule.

## Evidence Appendix:

No evidence pursuant to §§ 1.130, 1.131, or 1.132 or any other evidence has been entered by the Examiner and relied upon by appellant in the appeal.

Related Proceedings Appendix:

No prior or pending appeals, interferences or judicial proceedings are in existence which may be related to, directly affect or be directly affected by or have a bearing on the Board’s decision in this appeal. Accordingly, no copies of decisions rendered by a court or the Board are available.